1. What is a classification problem?

Prediction of the probability of each of the categories based on the given input variables/independent variable.

1. What is Logistic Regression?

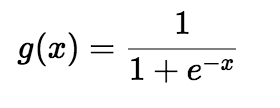
 Logistic regression is one such type of classification model which is used to classify the dependent variable into two or more classes or categories. It calculates the probability that a given value belongs to a specific class. If the probability is more than 50%, it assigns the value in that particular class else if the probability is less than 50%, the value is assigned to the other class. Therefore, we can say that logistic regression acts as a binary classifier.

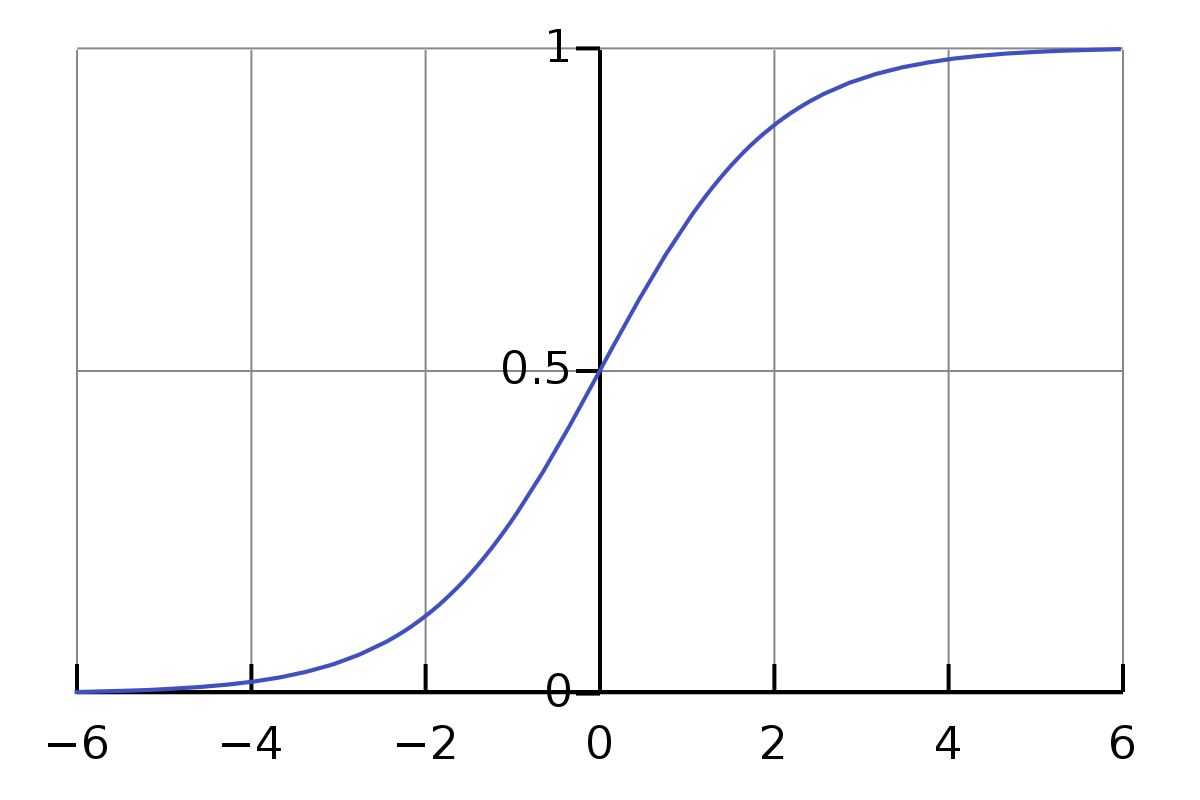
[**Types of logistic regression**](https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html#id19)

* Binary (Pass/Fail)
* Multi (Cats, Dogs, Sheep)
* Ordinal (Low, Medium, High)

1. Why is the Sigmoid function used for Logistic Regression?

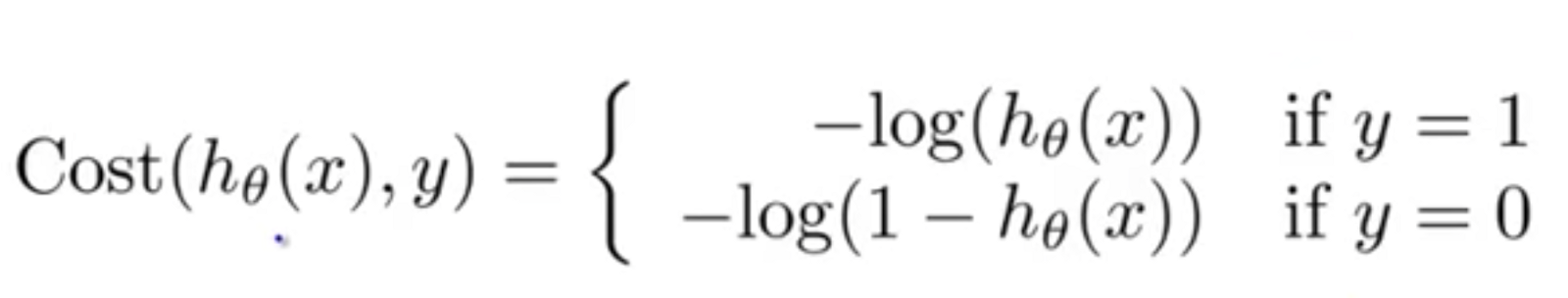
Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. We need the output of the algorithm to be class variable, i.e 0-no, 1-yes. Therefore, we are squashing the output of the linear equation into a range of [0,1]. To squash the predicted value between 0 and 1, we use the sigmoid function.



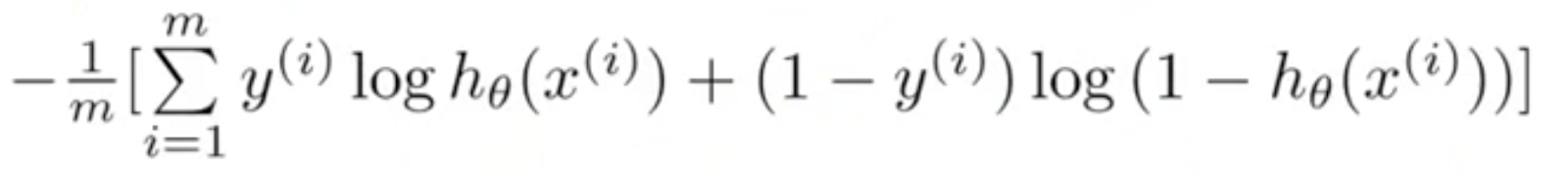


1. What is the cost function?

we use a logarithmic loss function to calculate the cost for misclassifying.



The above cost function can be rewritten as below since calculating gradients from the above equation is difficult.



1. What is multiple Logistic Function?

 simple logistic function for multiple features as:

Use multiple logistic regression when you have one [nominal](http://www.biostathandbook.com/variabletypes.html#nominal) and two or more [measurement variables](http://www.biostathandbook.com/variabletypes.html#measurement). The nominal variable is the dependent (Y) variable; you are studying the effect that the independent (X) variables have on the probability of obtaining a particular value of the dependent variable. For example, you might want to know the effect that blood pressure, age, and weight have on the probability that a person will have a heart attack in the next year.

1. What is multilabel Logistic Regression?

**multinomial logistic regression** is a [classification](https://en.wikipedia.org/wiki/Statistical_classification) method that generalizes [logistic regression](https://en.wikipedia.org/wiki/Logistic_regression) to [multiclass problems](https://en.wikipedia.org/wiki/Multiclass_classification), i.e. with more than two possible discrete outcomes.[[1]](https://en.wikipedia.org/wiki/Multinomial_logistic_regression#cite_note-1) That is, it is a model that is used to predict the probabilities of the different possible outcomes of a [categorically distributed](https://en.wikipedia.org/wiki/Categorical_distribution) [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable), given a set of [independent variables](https://en.wikipedia.org/wiki/Independent_variable) (which may be real-valued, binary-valued, categorical-valued, etc.).

1. How does the Logistic Regression Algorithm learn?

To minimize our cost, we use [Gradient Descent](https://ml-cheatsheet.readthedocs.io/en/latest/gradient_descent.html) just like before in [Linear Regression](https://ml-cheatsheet.readthedocs.io/en/latest/linear_regression.html). There are other more sophisticated optimization algorithms out there such as conjugate gradient like [BFGS](https://ml-cheatsheet.readthedocs.io/en/latest/optimizers.html#optimizers-lbfgs)

1. Explain the confusion matrix.



**True Positive:**

Interpretation: You predicted positive and it’s true.

You predicted that a woman is pregnant and she actually is.

**True Negative:**

Interpretation: You predicted negative and it’s true.

You predicted that a man is not pregnant and he actually is not.

**False Positive: (Type 1 Error)**

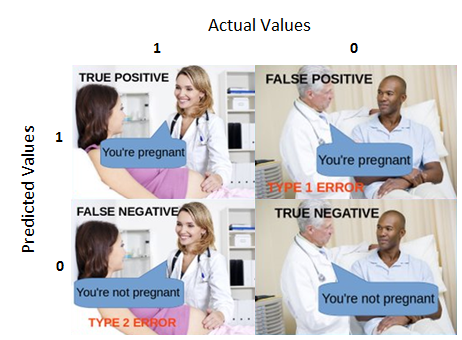
Interpretation: You predicted positive and it’s false.

You predicted that a man is pregnant but he actually is not.

**False Negative: (Type 2 Error)**

Interpretation: You predicted negative and it’s false.

You predicted that a woman is not pregnant but she actually is.



1. What is Accuracy?

**Accuracy** will be Out of all the classes, how much we predicted correctly, (TP+TN/Total cases)

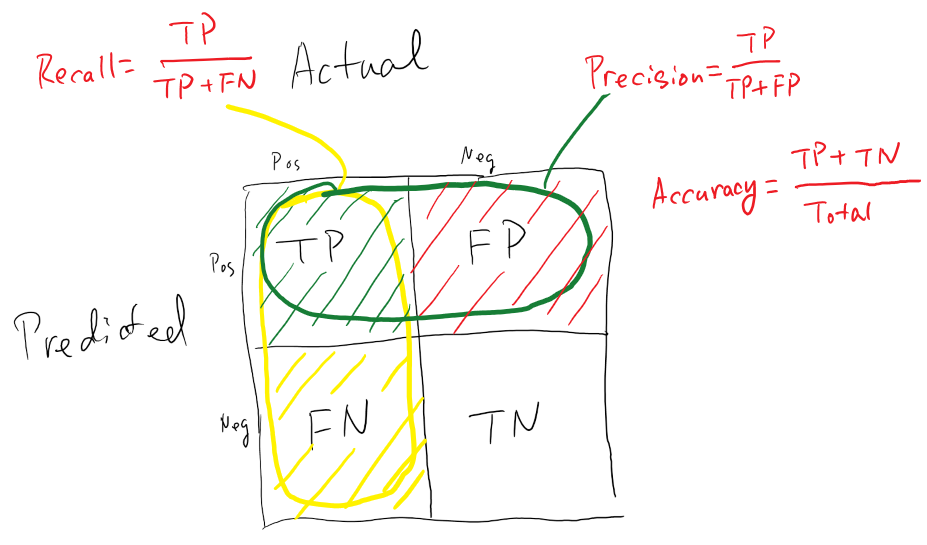
1. Why there is a need for other metrics if ‘accuracy’ is already present?

the model had a very high Accuracy but perform poorly in terms of Precision and Recall. So, necessarily Accuracy is not the metric to use for evaluating the model in this case.

1. What are Recall and Precision? How do they differ?

Recall :Out of all the positive classes, how much we predicted correctly. It should be high as possible.

Precision: Out of all the positive classes we have predicted correctly, how many are actually positive.

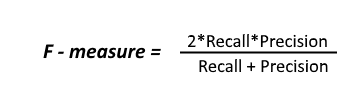


1. How does the tradeoff between recall and precision work?

with an increase in the Recall, there is a drop in Precision of the model.

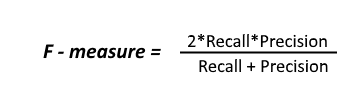
So the question is - what to go for? Precision or Recall?

It is difficult to compare two models with low precision and high recall or vice versa. So to make them comparable, we use F-Score. F-score helps to measure Recall and Precision at the same time. It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more.



1. Explain the F score.

F-score helps to measure Recall and Precision at the same time. It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more.



1. What is specificity? True Negative Rate

This represents how specific is the model while predicting the True Negatives. Mathematically,

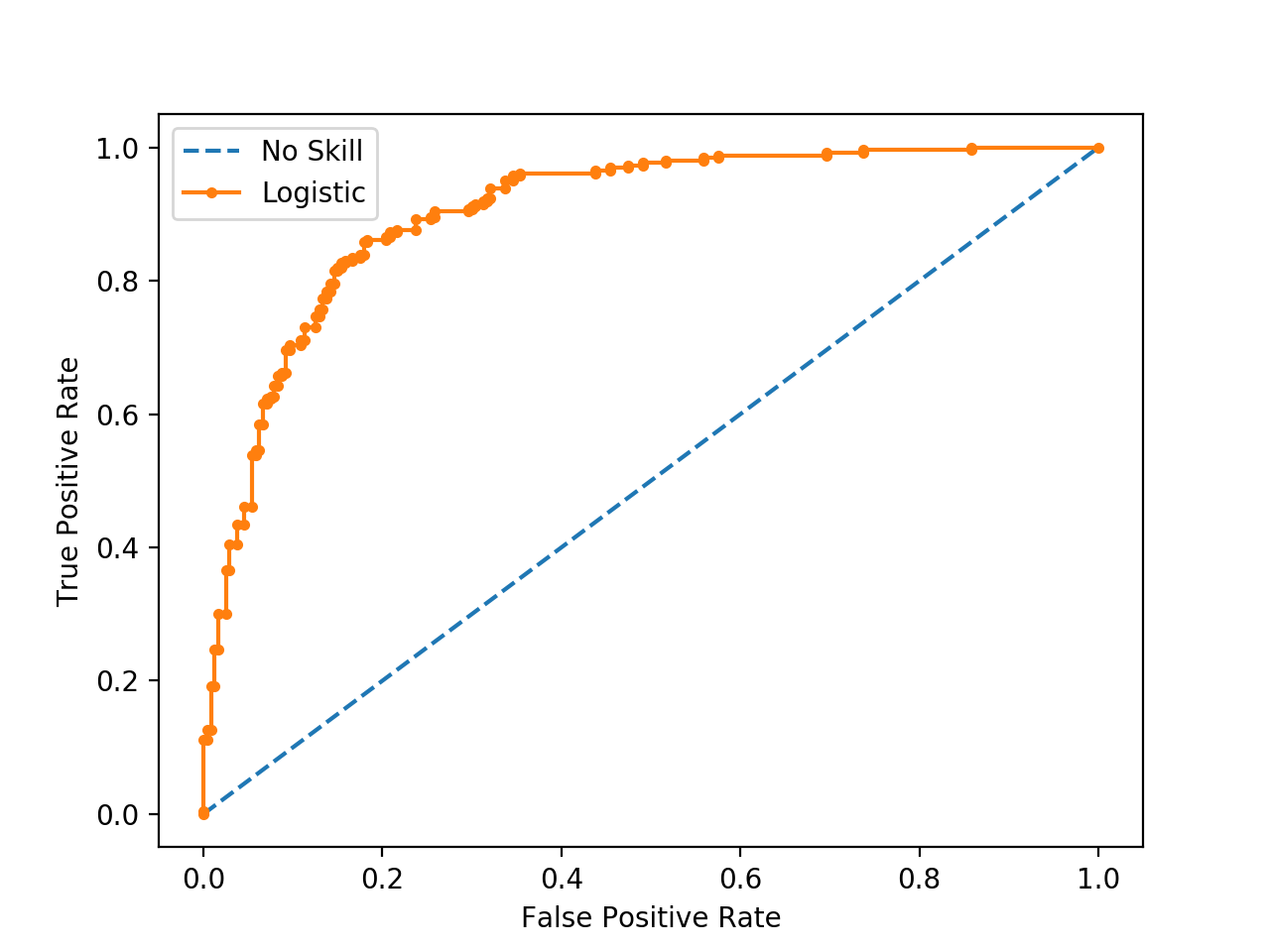
Specificity=𝑇𝑁(𝑇𝑁+𝐹𝑃)TN(TN+FP) Or, it can be said that it quantifies the total number of negatives predicted by the model with respect to the total number of actual negative or non favorable outcomes.

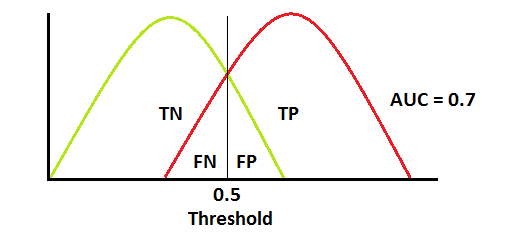
Similarly, False Positive rate can be defined as: (1- specificity) Or, 𝐹(𝑇𝑁+𝐹𝑃)

1. Explain the significance of ROC.

This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class.

An ROC curve is a commonly used way to **visualize the performance of a binary classifier**, meaning a classifier with two possible output classes.

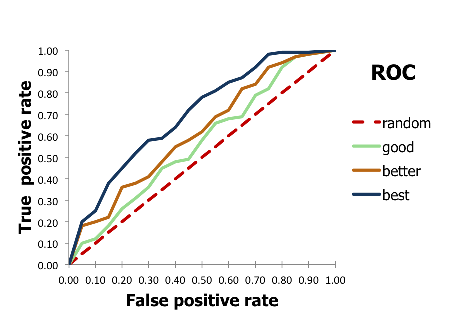




1. What is AUC, and when is it used?

When we have multiple models, this method helps to choose better model.

AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between patients with disease and no disease.

 <https://www.dataschool.io/roc-curves-and-auc-explained/>

<https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>

1. Explain the steps for Heroku deployment.
   1. After installing the Heroku CLI, Open a command prompt window and navigate to your project folder.
   2. Type the command **heroku login** to login to your heroku account.
   3. After logging in to Heroku, enter the command **heroku create** to create a heroku app. It will give you the URL of your Heroku app after successful creation. Or alternatively, you can go to the heroku website and create an app directly.
   4. Before deploying the code to the Heroku cloud, we need to commit the changes to the git repository.
   5. Type the command **git init** to initialize a local git repository.
   6. Enter the command **git status** to see the uncommitted changes.
   7. Enter the command **git add .** to add the uncommitted changes to the local repository.
   8. Enter the command **git commit -am "make it better"** to commit the changes to the local repository.
   9. Enter the command **git push heroku master** to push the code to the heroku cloud.
   10. After deployment, heroku gives you the URL to hit the web API.
   11. Once your application is deployed successfully, enter the command **heroku logs --tail** to see the logs.
2. What difficulties did you face while deploying to Heroku?

Nothing

1. Laying out notebook...

I've added these terms to the confusion matrix, and also added the row and column totals:



This is a list of rates that are often computed from a confusion matrix for a binary classifier:

* **Accuracy:** Overall, how often is the classifier correct?
  + (TP+TN)/total = (100+50)/165 = 0.91
* **Misclassification Rate:** Overall, how often is it wrong?
  + (FP+FN)/total = (10+5)/165 = 0.09
  + equivalent to 1 minus Accuracy
  + also known as "Error Rate"
* **True Positive Rate:** When it's actually yes, how often does it predict yes?
  + TP/actual yes = 100/105 = 0.95
  + also known as "Sensitivity" or "Recall"
* **False Positive Rate:** When it's actually no, how often does it predict yes?
  + FP/actual no = 10/60 = 0.17
* **True Negative Rate:** When it's actually no, how often does it predict no?
  + TN/actual no = 50/60 = 0.83
  + equivalent to 1 minus False Positive Rate
  + also known as "Specificity"
* **Precision:** When it predicts yes, how often is it correct?
  + TP/predicted yes = 100/110 = 0.91
* **Prevalence:** How often does the yes condition actually occur in our sample?
  + actual yes/total = 105/165 = 0.64
* **Null Error Rate:** This is how often you would be wrong if you always predicted the majority class. (In our example, the null error rate would be 60/165=0.36 because if you always predicted yes, you would only be wrong for the 60 "no" cases.) This can be a useful baseline metric to compare your classifier against. However, the best classifier for a particular application will sometimes have a higher error rate than the null error rate, as demonstrated by the [**Accuracy Paradox**](http://en.wikipedia.org/wiki/Accuracy_paradox).
* **Cohen's Kappa:** This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate. ([**More details about Cohen's Kappa.**](http://en.wikipedia.org/wiki/Cohen's_kappa))